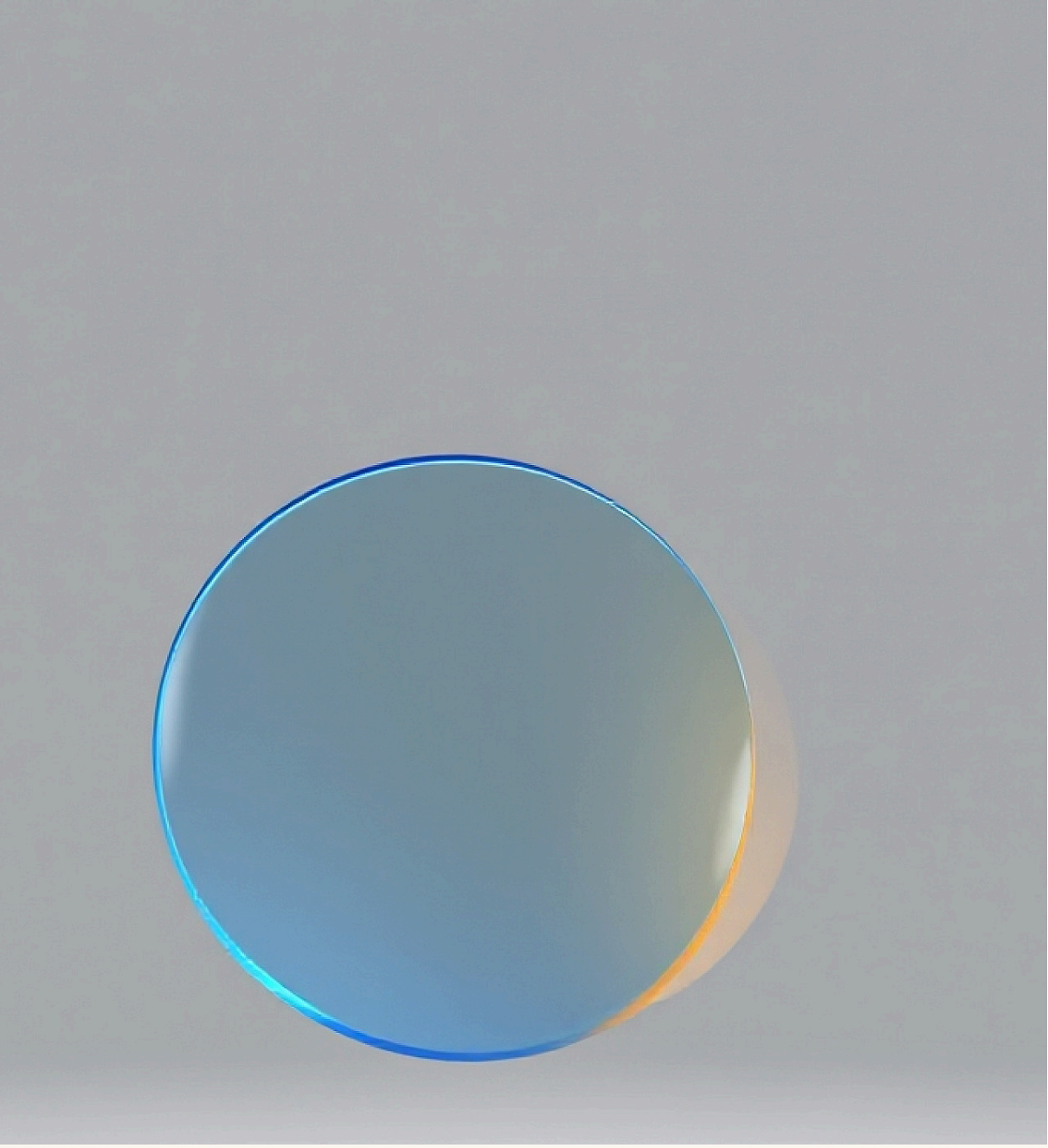
Kodlama ile programlama arasında bir fark var mı?



www.yzokul.com.tr

Kodlama ile programlama arasında bir fark var mı?



Kodlama ve programlama kelimelerini makalelerde, iş unvanlarında ve kurslarda görmüş olabilirsin. Kulağa aynı şeymiş gibi geliyorlar ve çoğu zaman insanlar onları aynıymış gibi kullanıyor. Yine de küçük bir fark var ve bunu bilmek her kelimenin gerçekte ne anlama geldiğini anlamana yardımcı olur. Kısa cevap şu: evet, bir fark var, ama bu tamamen ayrı etkinliklerin farkı değil, kapsam farkı.



Basit bir tanımla başla. Kodlama, bir bilgisayarın izleyebilmesi için Python veya JavaScript gibi bir programlama dilinde talimatlar yazmak anlamına gelir. Bu talimatlara kod denir ve bunları yazan kişi kodlama yapıyordur. Bu uygulamalı bir etkinliktir: bilgisayara ne yapacağını söyleyen kod satırlarını yazarsın. Kodlama, daha büyük bir işin bir parçasıdır, işin tamamı değil.



Programlama kod yazmaktan çok daha fazlasını kapsar. Baştan sona tüm süreç anlamına gelir: problemi anlamak, bir çözüm planlamak, kodu yazmak, test etmek ve sonradan düzeltmeler yapmak. Bunu bir projeyi baştan sona yönetmek gibi düşün. Kodlama, bu daha büyük çabanın yalnızca yazma kısmıdır. Bu yüzden programlamaya bazen yazılım geliştirme de denir.



Oklarla birbirine bağlanmış bir dizi kutu hayal et. İlk kutu problemi anlamaktır, ikincisi bir algoritma planlamaktır; algoritma, problemi çözmek için adım adım bir yöntemdir. Üçüncü kutu kodlamadır, dördüncüsü test etmektir, beşincisi ise bakımdır; yani programı zaman içinde çalışır durumda tutmaktır. Kodlama o dizideki yalnızca bir kutudur, dizinin tamamı değil.



Kodlama yalnızca yazma kısmıdır. Çözümü bir programlama dili kullanarak bilgisayara yazdığın yerdir. İki sayıyı toplayan veya ekranda bir mesaj gösteren birkaç satır yazabilirsin. Bu etkinlik odaklı ve somuttur: bilgisayarın çalıştırabileceği kod üretirsin. Ancak kodlama tek başına ne inşa edileceğine karar verir ne de çalışıp çalışmadığını kontrol eder.



Programlama, sorudan çalışan ve bakımı yapılan bir programa kadar tüm yolculuktur. Hangi problemi çözdüğünü sormakla başlar ve sen bitirdikten sonra da çalışmaya devam eden bir programla sona erer. Kodlama bu yol üzerindeki bir duraktır; önemli bir durak, ama yine de bir durak. Programcı planlar, yazar, test eder ve düzeltir; kodlayıcı ağırlıklı olarak yazar. Pratikte çoğu programcı ikisini de yapar.



Oyuncuların turları kazanarak puan kazandığı küçük bir oyun yapmak istediğini hayal et. Bu örnek, sonraki adımlarda farkı net biçimde gösterecek. Herhangi bir kod yazılmadan önce oyunun ne olduğuna, nasıl işlediğine ve oyuncunun ne yaşadığına karar vermelisin. Tüm bu düşünme ve planlama programlamanın parçasıdır ve kodlama başlamadan önce gelir.



Bir oyuncu kazandığında puan ekleyen kodu yazmak kodlamadır. Örneğin, her seferinde puanı on artıran bir satır yazabilirsin. O satır, bilgisayarın çalıştırabileceği küçük, somut bir yazı parçasıdır. İşin görünen, yazılan kısmıdır. İşte küçük bir örnek: score = score + 10. Bu, eylem halindeki kodlamadır.



Puanlamanın nasıl işlediğine ve kuralların ne olduğuna karar vermek programlamanın parçasıdır. Bir galibiyetin kaç puan değerinde olduğuna, bonuslar olup olmadığına ve oyunun ne zaman biteceğine karar vermelisin. Bu kararlar kodlamadan önce ve kodlamanın çevresinde alınır. Kodun ne yapacağını şekillendirirler. Net kurallar olmadan kodun yönü olmaz ve oyun iyi çalışamaz.



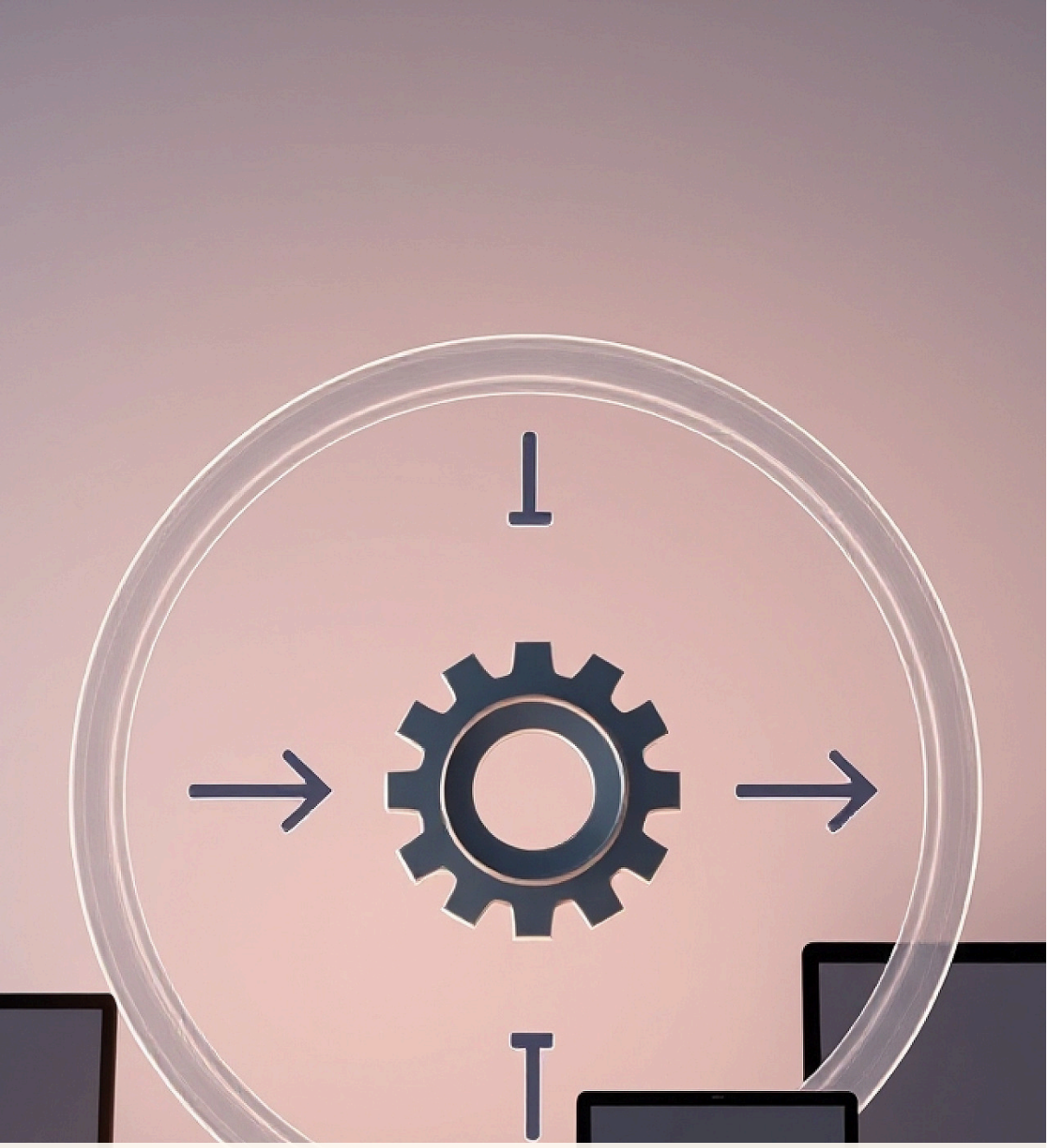
Puanların ve oyuncuların nasıl saklanacağına karar vermek de programlamadır. Her biri bir ad ve bir puana sahip oyunculardan oluşan bir liste veya basit bir tablo tutmayı seçebilirsin. Bu planlama veri yapısıyla ilgilidir; yani bilginin programda nasıl düzenlendiğiyle. Bu, kod yazmak değil planlamaktır. Buradaki iyi seçimler kodlamayı kolaylaştırır ve programı daha güvenilir kılar.



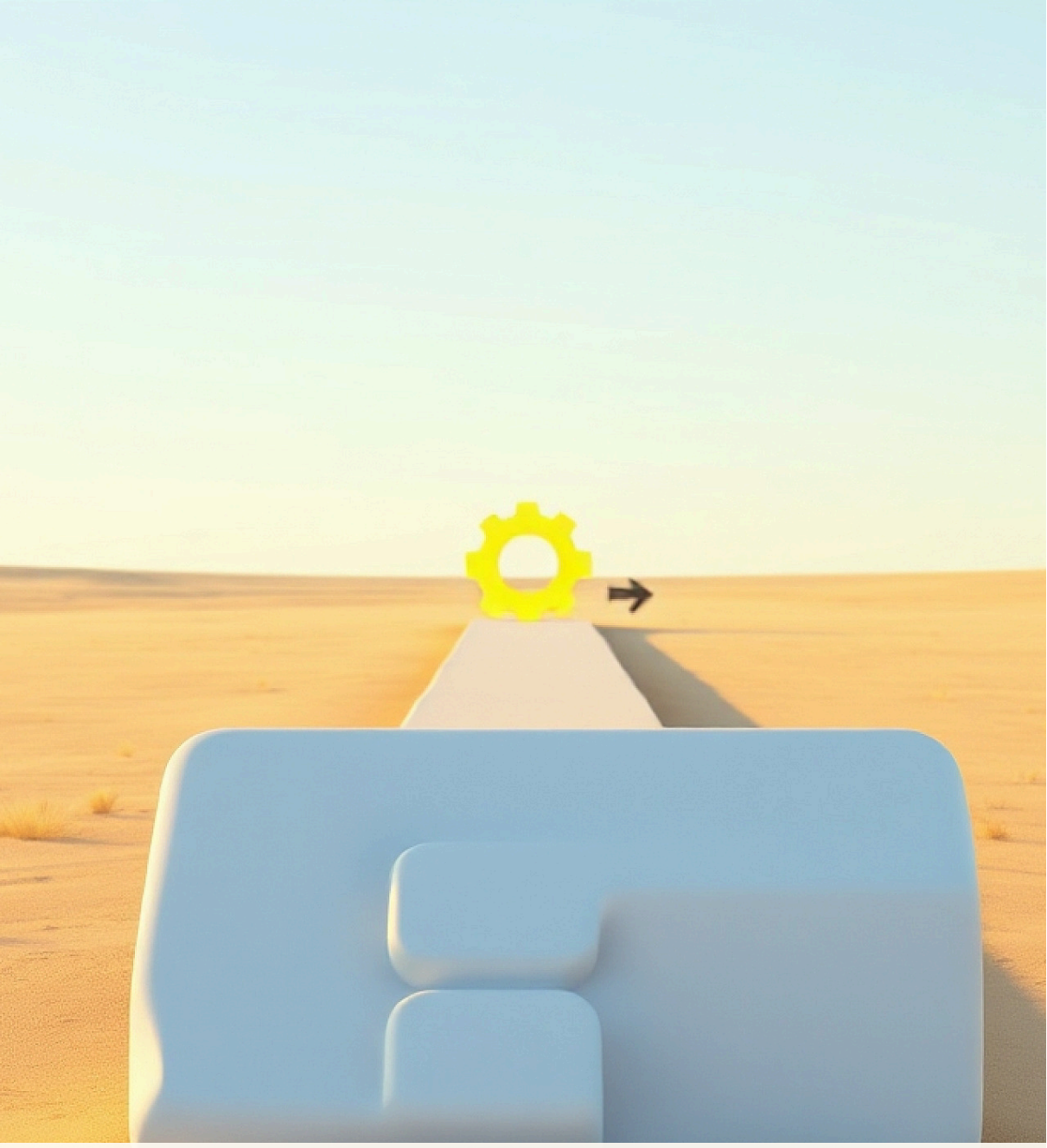
Oyunu denemek, hataları bulmak ve düzeltmek de programlamanın parçasıdır. Hata (bug), programın yanlış davranmasına neden olan koddaki bir yanlıştır. Oyunu çalıştırır, puanların yanlış olduğunu fark eder, sonra nedenini bulup onarırsın. Kodlama tek başına işi bitirmez. Test etmek ve düzeltmek tüm sürecin vazgeçilmez parçalarıdır ve çoğu zaman kod yazmak kadar zaman alır.



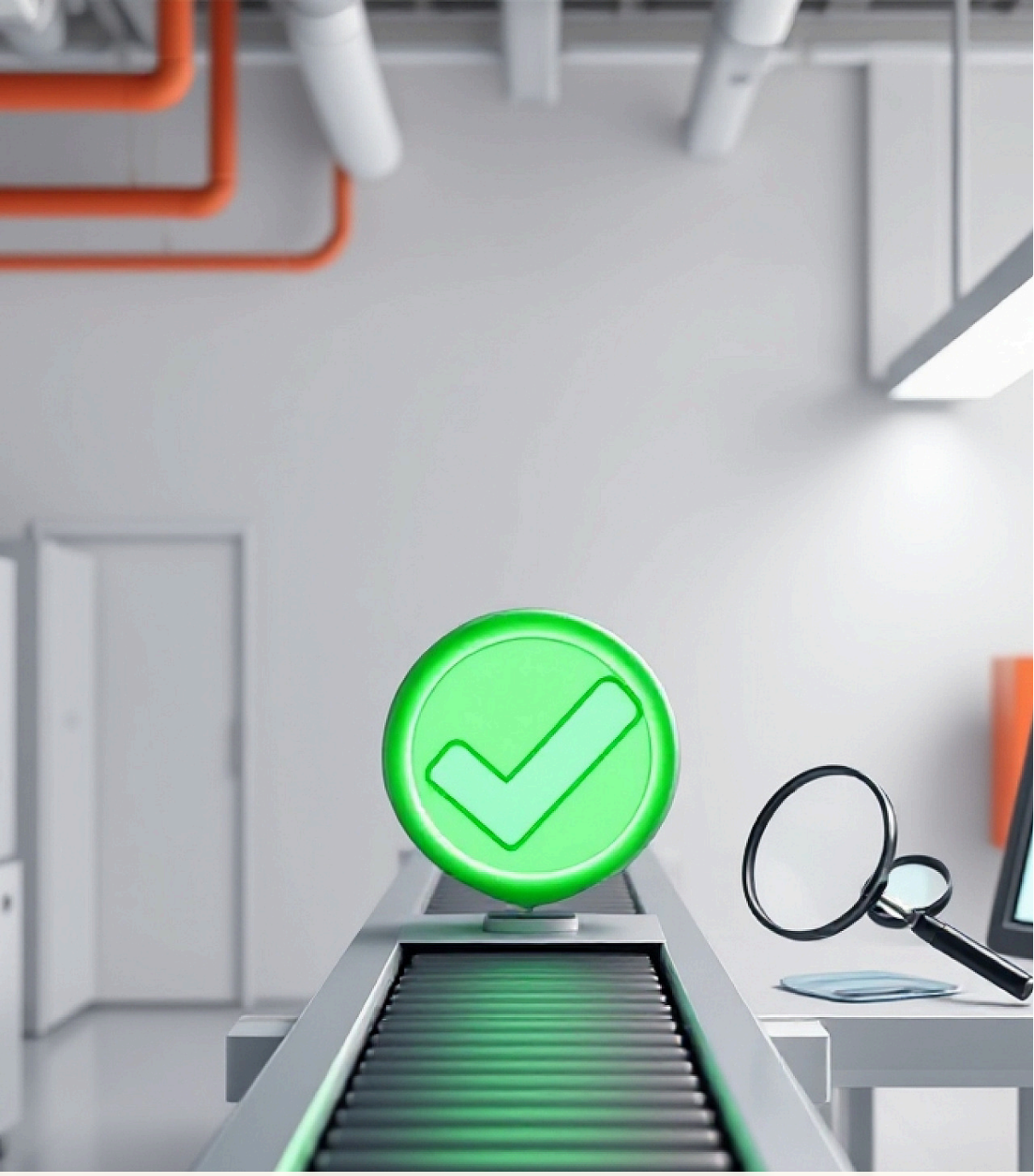
Normal konuşmada insanlar kodlama ve programlama ile aynı şeyi kasteder. Birinin kodlama yaptığını söylediğini duyarsın ama aslında tüm süreci kastediyordur ve bu yaygındır. Bu gündelik kullanım yanlış değildir; sadece daha az kesindir. Günlük konuşmada iki kelime çoğu zaman aynı anlamda kullanılır ve herkes ne kastedildiğini anlar.



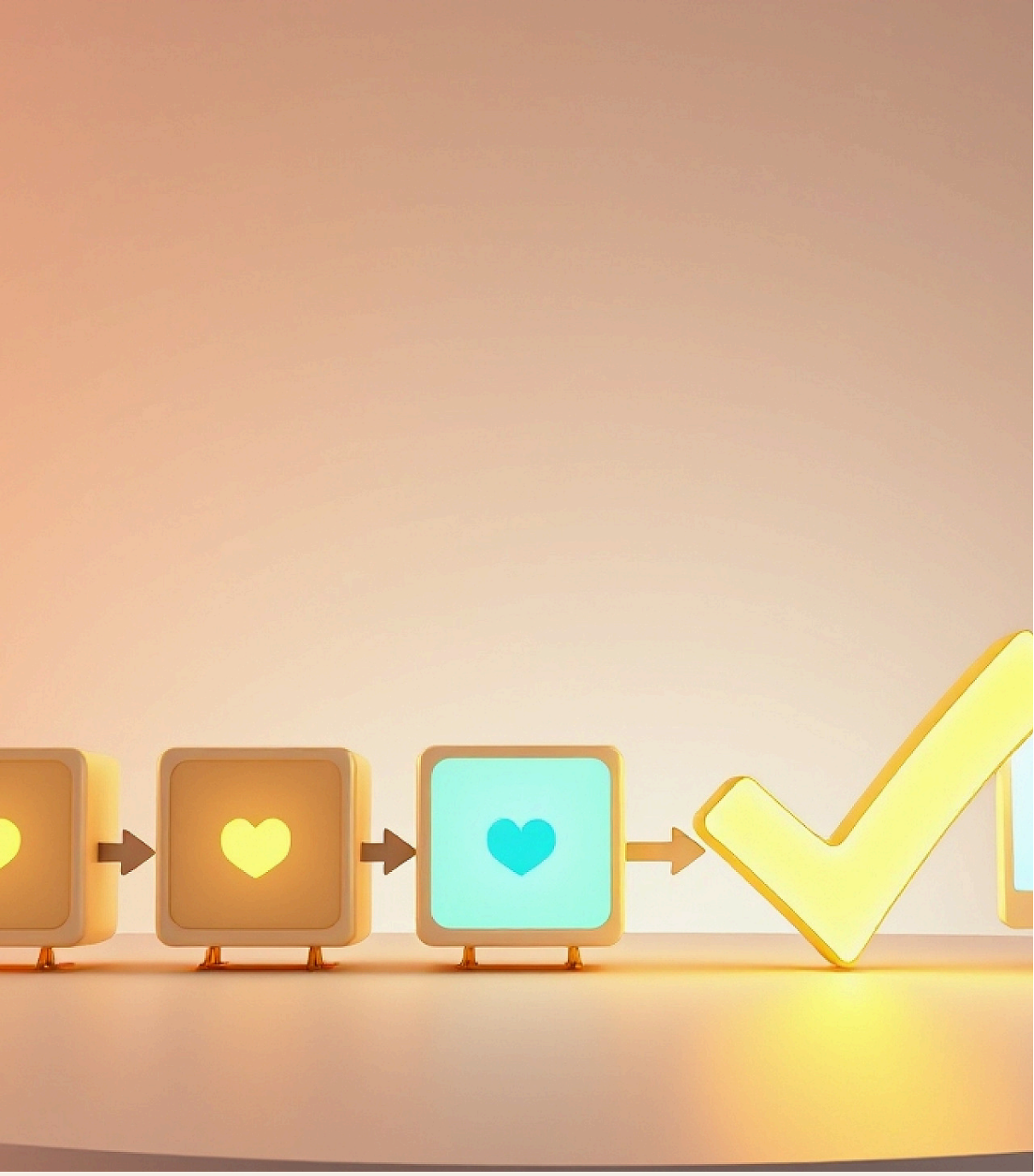
Kodlama ile programlamanın tamamen farklı şeyler olduğunu düşünme hatasına düşme. Ayrı dünyalar değillerdir; kodlama programlamanın içindedir. Programlama, kodlamayı adımlarından biri olarak içerir. Yani her kodlayıcı, planlama veya test yapmasa bile programlamanın bir parçasını yapıyordur. İkisi bağlantılıdır, karşıt değildir ve biri diğerini içerir.



Doğrudan kod yazmaya atlama hatasından kaçın. İyi programlama önce problemi anlamakla başlar. Planlamayı atlarsan yanlış problem için kod yazabilir veya dağınık bir çözüm oluşturabilirsin. Programın ne yapması gerektiğini ve nasıl çalışması gerektiğini sormak için zaman ayır. O zaman kodlama daha kolay ve daha odaklı hale gelir ve daha az emek harcarsın.



İşin kod yazıldığında bittiğini düşünme. Test etmek ve düzeltmek programlamanın parçasıdır ve kod yazmak kadar önemlidir. Hiç test edilmeyen bir program, insanlar kullandığında başarısız olabilir. Programını denemek, hataları bulmak ve onarmak için zaman planla. Programı zaman içinde çalışır durumda tutmak olan bakım da sürecin parçasıdır.



Özetle: kodlama, çözümü bir programlama dilinde yazmaktır ve programlama, onun etrafındaki tüm süreçtir. Kodlamanın programlamanın bir parçası olduğunu, ayrı bir etkinlik olmadığını unutma. Günlük konuşmada iki kelime çoğu zaman karıştırılır ve bu sorun değildir. Net iletişim için yazma işi için kodlama, tüm yolculuk için programlama kullan.